



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Managing Service Development (SaaS) as a Project:

Business Process Modeling

MSc in Project Management and Product Development

by

University of Piraeus

Vasiliki Iliadi

Supervising Professor: Dimitrios Emiris

Project Management Dpt of Industrial Management and Technology, 2017

ΔΗΛΩΣΗ

«Η εργασία αυτή είναι πρωτότυπη και εκπονήθηκε αποκλειστικά και μόνο για την απόκτηση του συγκεκριμένου μεταπτυχιακού τίτλου».

«Τα πνευματικά δικαιώματα χρησιμοποίησης του μη πρωτότυπου υλικού ΜΔΕ ανήκουν στο μεταπτυχιακό φοιτητή και το επιβλέπον μέλος ΔΕΠ εις ολόκληρο, δηλαδή εκάτερος μπορεί να κάνει χρήση αυτών χωρίς τη συναίνεση άλλου. Τα πνευματικά δικαιώματα χρησιμοποίησης του πρωτότυπου μέρους ΜΔΕ ανήκουν στον μεταπτυχιακό φοιτητή και τον επιβλέποντα από κοινού, δηλαδή δεν μπορεί ο ένας από τους δύο να κάνει χρήση αυτού χωρίς τη συναίνεση του άλλου. Κατ' εξαίρεση, επιτρέπεται η δημοσίευση του πρωτότυπου μέρους της διπλωματικής εργασίας σε επιστημονικό περιοδικό ή πρακτικά συνεδρίου από τον ένα εκ των δύο, με την προϋπόθεση ότι αναφέρονται τα ονόματα και των δύο (ή των τριών σε περίπτωση συνεπιβλέποντα) ως συν-συγγραφέων. Στην περίπτωση αυτή προηγείται γραπτή ενημέρωση του μη συμμετέχοντα στη συγγραφή του επιστημονικού άρθρου. Δεν επιτρέπεται η κατά οποιοδήποτε τρόπο δημοσιοποίηση υλικού το οποίο έχει δηλωθεί εγγράφως ως απόρρητο».

Abstract

In the context of the present thesis, we will be studying core principles of Business Process Management, and how we can take advantage of them in combination with Project Management Methodologies and modeling tools in the context of Software as a Service businesses and their development.

Initially we provide the reader with an introduction to Business Process Management, how it can be used, and how the life cycle is structured. We further define the first three phases of the life cycle to understand how this process is kickstarted, and how it can assist forward thinking and planning for a business that has not yet been created. We continue to present a short history of software in the market, from on premise to SaaS, and introduce the SaaS model in detail for the reader.

In the next stage we move on to project management practices that can be used to manage the development of a SaaS service, and which methodologies are best for that business model. We continue by introducing the maturity factor of a SaaS business by presenting two maturity models that could represent a SaaS business' growth phases.

Following this, we detect five distinct stages and set up a more relevant maturity model for SaaS businesses, combining information from the two theories presented before as well as utilizing personal experience.

Lastly, we implement this maturity model for a SaaS business to capture its growth utilizing Business Process Model and Notation and swim lanes, showcasing the design and evolution of processes through the model's phases.

Acknowledgements

Before wrapping up my studies for a few years, and before delving into a strong professional future, I was able to complete this thesis after long hours of research after work.

Without my mother Matina, father Kostas and brother Petros I could not have done it.

Thank you.

Vasiliki Iliadi

1/10/2017

Contents

1. Business Process Management	v
1.1 Introduction	1
1.2 What is a process?	2
1.3 Business Process Management Life Cycle	4
1.4 Identification, Discovery and Analysis	6
1.5 Benefits	9
2. Software as a Service	12
2.1 On-premise software	12
2.2 Cloud Computing	13
2.3 Software as a Service	15
2.3.1 SaaS Model Analysis	17
2.3.2 Characteristics and Benefits	19
3. Managing SaaS	22
3.1 Project Management according to PMI	22
3.2 Project Management Methodologies	23
3.2.1. Traditional Project Management with Waterfall	23
3.2.2 Agile	24
3.3 Scrum	27
4. Maturity of a SaaS business	32
4.1 The Business - IT Maturity Model	32
4.2 IT Infrastructure and Operations Maturity Model	35
5. Case Study	38
5.1 Tools	38
5.2 Maturity Model and Analysis	38
6. Conclusion	50
7. References	52

1. Business Process Management

1.1 Introduction

Every type of organization - be it a small shop in the corner of your neighborhood, or a massive holding company like Alphabet - has one thing in common. It is compiled by actionable events and activities; processes. While the number of processes may be different depending on where a company is in the spectrum, these processes are what makes up the core value chain that can make the company successful. A couple of examples, Ikea's - likely close to thousands - processes on logistics management, combined with their business logic on being accessible and focusing on product development, has been refined so much so there is barely any waste, and each step is efficient. Comparatively, the shop just across your door has to keep track of their processes, like bringing in new merchandise, processing an order and giving back change. The results of the processes are what makes everything work after all; something is done according to planned steps, and a "transaction" is completed successfully.

The way a company handles processes - designs and performs - directly affect the efficiency and the quality of what is provided; from a service to a product deliverable. Because of the importance this section carries, a discipline has been created with the sole purpose of analyzing and optimizing what makes this successful, Business Process Management.

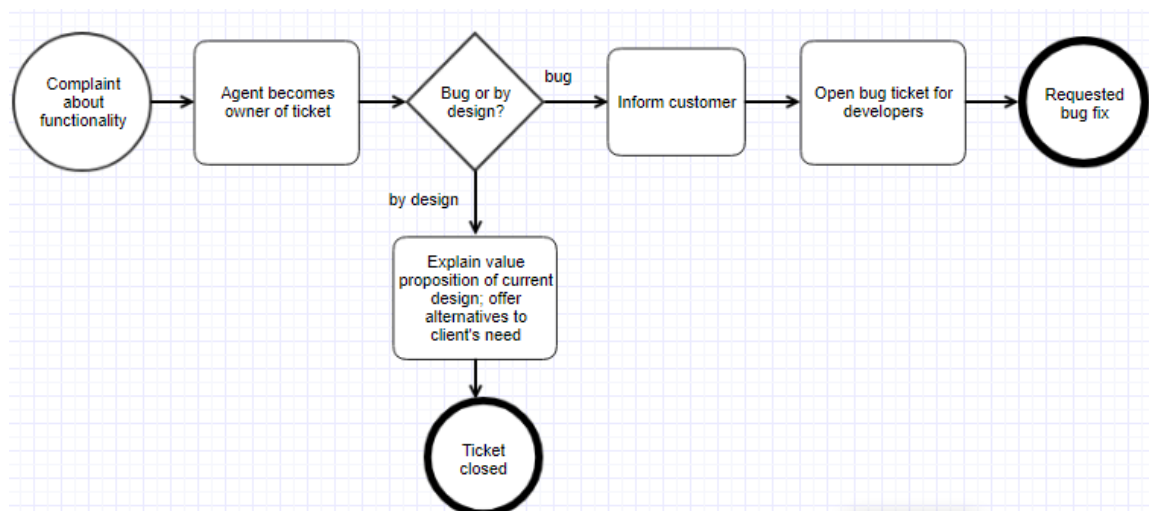
According to Gartner, Business Process Management is "the discipline of managing processes (rather than tasks) as the means for improving business performance outcomes and operational agility" (Gartner, 2017). Gartner also points out that processes

link together what the Project Management Institute (PMI) defines resources for a project (people, information, machinery etc.) to generate value for the customers and the company itself, throughout the organization. BPM is usually part of operations management, and essentially enables process optimization, starting from detecting the processes in an organization and designing upcoming ones (design), though optimization and reengineering when the need is present.

We will be going through a short journey of the full circle of BPM to create a common understanding and provide definitions about the basic principles, and will later focus on Business Process Documentation, Business Process Mapping and Rapid Process Discovery which will be used further down the line in our empirical analysis.

1.2 What is a process?

To be able to see an overview of the life cycle of BPM, it is vital to establish a common understanding of what a process is made up of. Let's go through an example of a customer support agent dealing with a customer complaint about a possible defect in a software in a high-level chart. The example process is presented in Diagram 1.1 below.



1.1 Example: Handling a customer complaint

In this example process we have the incoming complaint from a user in regard to a software tool that is being handled by the customer support team. Originally, we have the incoming event - the complaint. In the next step mapped, we can see the complaint is assigned to a member of the customer support team to handle - it is now in their hands to review the ticket - represented by a task, as it is a simple procedure - and deduct which case is true; does the complaint represent a valid issue in the software, or is the client misusing a feature to achieve a goal and not getting results? This is shown in the gateway node; depending on the case, two outcomes are possible. If the complaint is valid, then the agent informs the customer that the team will be investigating and preparing a fix, while also notifying the developers about the issue so it can be worked on.

In regard to the second case from the gateway node, which occurs fairly often, a client is trying to achieve something but is not using a specific feature the intended way. Many a times, a customer support agent can guide the user by presenting the original intended way the feature was designed for, and can usually offer an alternative way for the client to achieve the end goal while also providing info for the system's value proposition.

For this specific process, we opted to only show the process up until the ticket is created just to showcase how process mapping works; if we were to create a process about how to manage a bug fix, or expand this one to include it, the mapping would have further activities about resolving the issue. Deploying it in production and notifying the client back that the bug is resolved. Based on what the need is, more complicated processes can include many more ingredients to fully map detailed steps of more complicated cases. For example, in case of more complicated processes it is useful to depict the actors as well - here we have the customer and the support team - as well as any physical objects,

tools or records - perhaps the program customer support is using, and any relevant documents linked about how conversations are assigned to agents.

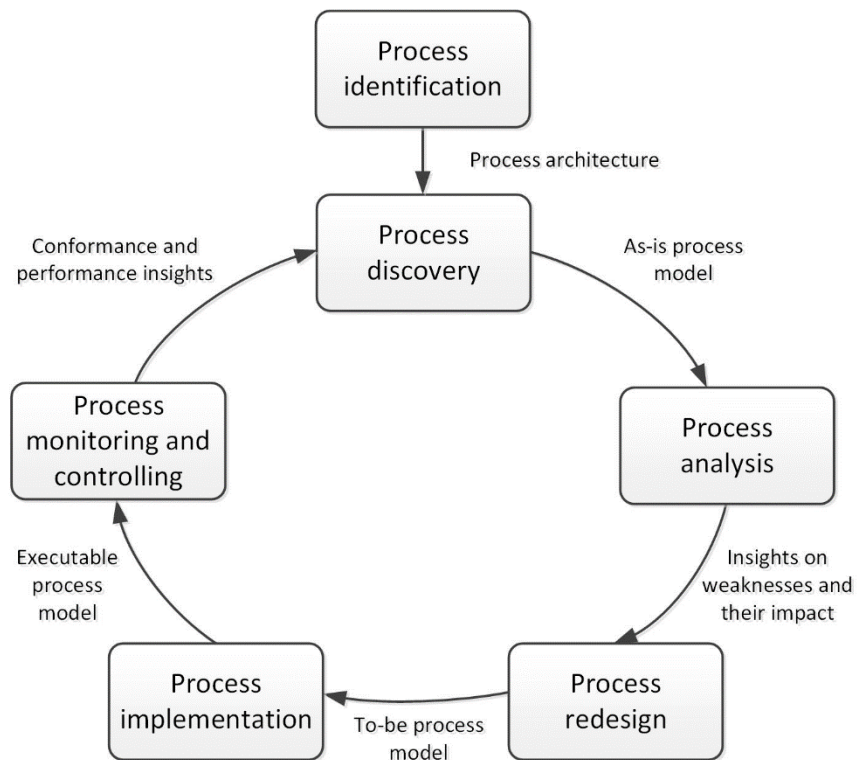
Overall, the execution of the process leads to an outcome, that usually has a value for the customer directly or to the company itself. Here in both outcomes, the customer receives insight relevant to their complaint.

1.3 Business Process Management Life Cycle

With a good understanding of the process thinking ideology and what it represents, one can see BPM as a cycle composed of the following stages (Dumas et al., 2013):

- **Process Identification:** For an existing business wanting to make BPM an integral part of their operation, this is the first step to success. Problems are presented so relevant processes that can assist in resolving it can be detected, to gain further focus down the line. Here metrics are also set, to enable the company to realize what is important and needs to be improved, and to allow them to quantify any later progress. This step is also very important in passing a more process oriented view throughout the company.
- **Process Discovery and Process Analysis:** Here, current processes are being identified, documented in detail and are analyzed to detect possible issues affecting performance, cost or other factors of the organization.
- **Process redesign/process improvement:** After issues have been detected, this phase allows the team to identify changes that can be implemented/embedded within the current processes to improve the goals of the company and possibly resolve the issues identified in the first phase.

- Process implementation: in this stage, the changes decided and finalized upon from the previous phase are prepared and performed.
- Process monitoring and controlling: in this stage, based on the metrics the company has selected to track, data is gathered and analyzed to detect whether the process changes have brought improvement. If not, the cycle needs to be repeated.



1.2 BPMN Life Cycle

In simple words, BPM asks the following questions:

- What are we doing?
- How efficient is what we are doing?
- Why are we not more efficient?
- What can we change to become more efficient?

While this completes the representation of the BPM journey in a company, for this paper we will be focusing on 3 of the said phases: Process Identification, Process Discovery and Process Analysis. This will be analyzed in the context of a SaaS company just starting out, and will be mapped in process diagrams at a later point in the paper.

1.4 Identification, Discovery and Analysis

As we mentioned before, the process that kickstarts Business Process Management is Process Identification, which allows the organization to clearly define the processes through departments and roles, and set criteria to prioritize them. While the ideal would be for all processes to be documented in full detail, this is usually costly and does not pay back the effort of recording everything. Therefore, the organization would need to prioritize in which processes it would be wise to focus on, and then further analyze deeper within them. Usually, in established organizations that decide to apply the BPM principles, a few processes are already standing out as they are obviously connected with problems the organization faces and would like to resolve through this change management effort.

Another vital part for Process Identification is agreeing on using a categorization, something that will significantly help in enumerating the processes as well as locating the boundaries between each one. One of the most widespread options is following Michael Porter's Value Chain, dictating that there are very few processes within an organization (Dumas et al., 2013). Porter distinguishes two separate categories of processes: core processes, that include the activities that add value to whatever the client pays for (sales, materials, logistics etc.) and supporting processes, that enable the execution of core processes but do not directly add value to the product. Literature also lists a third category

including management processes, but when it comes to such categorization each organization can optimize it to opt for what works for their individual needs. At any case, the listing of processes should always have in mind their importance as well as the complexity in managing them every day.

Moving on to Process Discovery; here's where an organization gathers all the necessary information to create precise as-is models of their existing processes. Here would be the moment where a process analyst would utilize a modeling language to depict what the company does in everyday operation to an easy to communicate and improve on model - and do this for the processes prioritized by the Process Identification. The analyst would need to have domain knowledge, as well as have all the necessary information to properly create the diagrams.

While a widespread language for diagram models is simple flowcharts, there are more detailed languages for process mapping like Unified Modeling Language and Business Process Model and Notation. Unified Modeling Language (UML) is a standardized, general-purpose modeling language in the field of software engineering developed in 1990s, and accepted by the International Organization for Standardization (ISO) as industry standard for modeling software intensive systems in 2000. Out of the 14 types of diagrams available in UML, the activity diagram is suited for business process modeling.

While very similar to UML, Business Process Model and Notation (BPMN) is a graphical representation for specifying business processes in a business process model created in 2004 (Bpmn.org, 2017). While UML has an object-oriented approach when modeling applications and a focus on software design, BPMN takes on a process oriented approach which is more suitable in a business process domain. Due to this, BPMN is

becoming the go-to modeling language for business process modeling initiatives. Utilizing BPMN can be simple or complicated, based on the process mapped and the need requiring mapping - for example, the process in Figure 1.1 is utilizing BPMN, the main elements of which are tasks and decision nodes. Based on the analysis and the level of detail, more symbols can be engaged in the modeling representing mainly activities, events, control nodes, flows and tasks. Often, further documentation is required to be included in a process depiction as well, and based on the level of detail the company has decided to delve into, each process can be enriched with further activities and text annotations to go as close to “as-is” as possible.

Once a company gets their hands on mapping out their as-is processes, they can investigate what goes wrong in their current regime in the Process Analysis phase. This is the step to detect possible weaknesses of the current model, and analyze in detail what their effect is on the company. Weaknesses are not only delays in the process - they can be as simple as having a single person blocking being responsible for a specific task, causing them to be a blocker for any dependent processes, or it could be a detail during the hand off of a project from one team to another that could potentially prevent confusion for both teams. It also brings to the surface simple tasks that could be automated to enable employees to work on what’s important, and not with tedious tasks. Going into a BPM initiative, it is usually the case where participants already have some issues in mind that can possibly be optimized, or there are some major, easily detectable issues that can be refined and lessen related costs.

In the context of our thesis, we will be looking at things zooming out. As we describe in detail further down the line, we investigate beneficial a process discovery analysis can be to a SaaS company even before its creation. How can proactive process

analysis, based on usual happy paths throughout different business development stages, assist in a company's long-term planning - or even keep it at the road to success? What bullets can be dodged, and what can we already see coming, to properly prepare to face it?

1.5 Benefits

The reason we map processes is because they are clear in telling us what to do - and that can become evident throughout the BPM life cycle in multiple ways, as it brings to the table multiple benefits:

1. **Transparency:** Whatever takes place is written down in detail, each person knows exactly what they are responsible for, and new employees pick up the pace by locating where they fit in each process easily. Uncertainty is eliminated.
2. **Communication and coordination:** Processes are a cross-organization tool that can be used as a common language between stakeholders; a map can facilitate communication in ways a meeting with unclear activities between two departments never could. Developers and marketing can now coordinate just by glimpsing at the same swimlane for each release.
3. **Quality:** Processes are one of the most important tools for quality control; they ensure and standardize the consistency and quality of the outputs for each activity.
4. **Elimination of an individual function:** In a SaaS company there is really no individual function, as all departments support and need each other. Process discovery eliminates a team focusing on the efficiency of the team, but switches towards the efficiency of end to end processes.
5. **Knowledge management:** Everything is recorded - not everything is dependent over a single person, but knowledge is distributed to the team, eliminating

unnecessary blockers and waiting times. New hires can get up to speed in significantly less time with all the documentation available.

6. Predictability: Analyzing as is processes with a future vision allows for early additions of corrective changes to ease in changes and allow scalability. While impossible to be 100% accurate, filtering the analysis through the company's vision can ease in necessary changes for the future.
7. Optimization: These three processes kickstart the cycle for BPM, opting for increase in efficiency, reduction of costs, easier facilitation of risk management and constant steps towards optimized processes.

Later, we will be able to see how this can become evident during empirical analysis.

Cultivating Process Culture within an organization

Giving a go in trying BPM does not necessarily guarantee that it will work; it should be embedded in the culture of people to help retaining it and allowing it to scale along with the organization. In simple words, this is an effort to have people work smarter and not harder, with often the same resources.

The most important thing is to cultivate an open culture and train for change management from the beginning; for a startup just kicking off, it is vital for everyone to be aware and familiar with the fact that nothing will stay the same and changes will constantly come up. For this, cultivating process culture imprints employees with the need to work not just to maintain, but to improve their processes, as well as keep an up to date documentation about how they do their work; many organic changes that could be overlooked otherwise might significantly affect an output of a specific process.

Incorporating the task of “updating documentation” into the task relevant to the documentation is a game changer. With an up-to date documentation throughout the company, each Plan-Do-Check-Act cycle in BPM can only become easier as you can track the differences and reflect on your Key Performance Indicators (KPIs) to see what works and what doesn't.

Engaging the employees with the processes and documentation itself also engages them with the company's success - supporting the core processes of somebody's work and engaging them to make it better is key to cultivate innovation, as you are able to define “the box” and then think “outside” of it.

Most importantly - if possible, start applying bits of process culture from scratch.

2. Software as a Service

2.1 On-premise software

Back in the early 1970s, applications were created by Oracle, SAP and other software giants using the state of the art technology and selling to the clients that could afford it; the enterprise sector. Back then, state of the art software translated into software running on the company's own servers, requiring installation and intricate maintenance by specialists. While costly, the technology was at the time paying itself as it saved a company time and money. To run this software, the enterprise usually had to install it to their own servers (or buy servers for this purpose), set up implementation operations and maintenance tasks and pay for training, support and updates - and it was popular, with strong examples like SAP's ERP and Siebel's Customer Relationship Management tool.

This is type of tool is the on premises software, and typically requires a license for every server or every end user within the company. From a technical aspect, an on-premise software allows any customization required in its architecture as each machine gets its own version of software to run - so any customizations can be implemented and adjusted based on a company's needs.

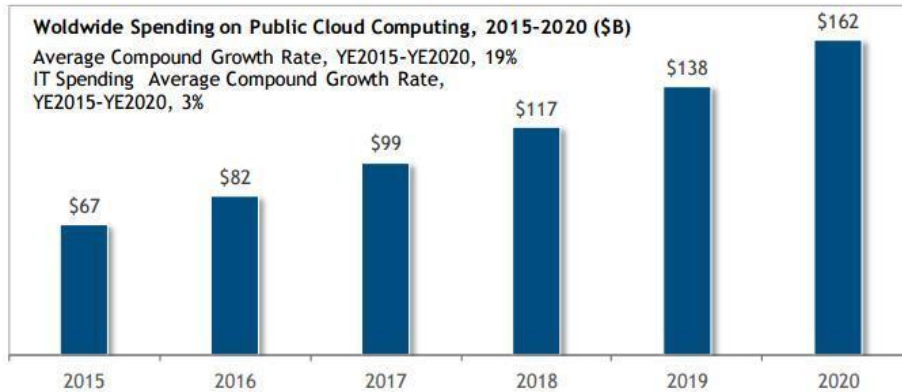
While this solution was robust and customizable, it was hard to make it scalable; the cost of hardware, support and maintenance, along with lack of accessibility as everything is within the company's building weakened the benefits of customization - especially as a competitive technology slowly emerged.

2.2 Cloud Computing

With the rise of Web 2.0 and what came to be the standardization of the internet not only to every business - Small and Medium Businesses(SMBs) to Enterprise - but to our homes, many doors opened to allow software to generate what is called cloud computing. The cloud computing paradigm allows access to configurable resources accessed over the Internet (Buyya et al., 2009). Simply put, someone can use someone else's resources to store and process data. Besides the ease of accessing these resources, management is usually also easy as you can adjust the resources you need with a few clicks at any moment. For a business this can be storage (Google Drive), infrastructure (Heroku), servers (Amazon Web Services), applications (Mailgun) and any other solutions required to deliver a software product. But the cloud is not only utilized for business-to-business (B2B) tools, which has to do with why it has become so popular; many tools are offered for consumers - services like Netflix for watching on demand shows, Spotify for music and even Google Drive for consumers has a widespread use and keeps growing. For the context of this thesis, we will be focusing on B2B tools.

Cloud computing did not only spread because of reducing the cost a company had to endure for hardware, licensed software and maintenance - it's easy to set up, manage and customize, opens access to a service globally and device-wide, increases the market and enables teams to adjust resources as needed as opposed to hosted applications. It provides scalability and simplicity at a low cost, as well as high performance, allowing it to continuously gain popularity since the 2000s. According to statistics provided by International Data Corporation (IDC), the worldwide spending on cloud computing will be reaching \$162 billion by 2020 (Salesforce/IDC, 2016).

The Rapid Growth of Cloud Computing, 2015-2020



Source: IDC, 2016

2.1 IDC Whitepaper: The Salesforce Economy: Enabling 1.9 Million New Jobs and \$389 Billion in New Revenue Over the Next Five Years, 2016

From a technical point of view, cloud computing consists of one or more clients, which are the devices that will access the cloud solution (PCs, mobile, laptops etc.), data centers and distributed services.



2.2 Cloud Computing Architecture

As cloud computing can refer to any tool and any audience, there are multiple delivery models;

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- Security as a service (SECaaS)
- Mobile "backend" as a service (MBaaS).

For the context of this thesis, focus will remain on the SaaS delivery model.

2.3 Software as a Service

Perhaps the most widely spread delivery model of cloud computing is Software as a Service, and likely has to do with it being one of the main delivery models able to go for both B2B (Business to Business) as well as B2C service (Business to Consumer). From Netflix and Spotify to Google Business and Office 365, SaaS applications are cloud services that deliver software to users over the Internet. As seen below, according to Eurostat's survey for 2016, many of the categories of cloud computing most used by enterprises falls under SaaS tools, like office software, financial or accounting software and Customer Relationship Management Tools.

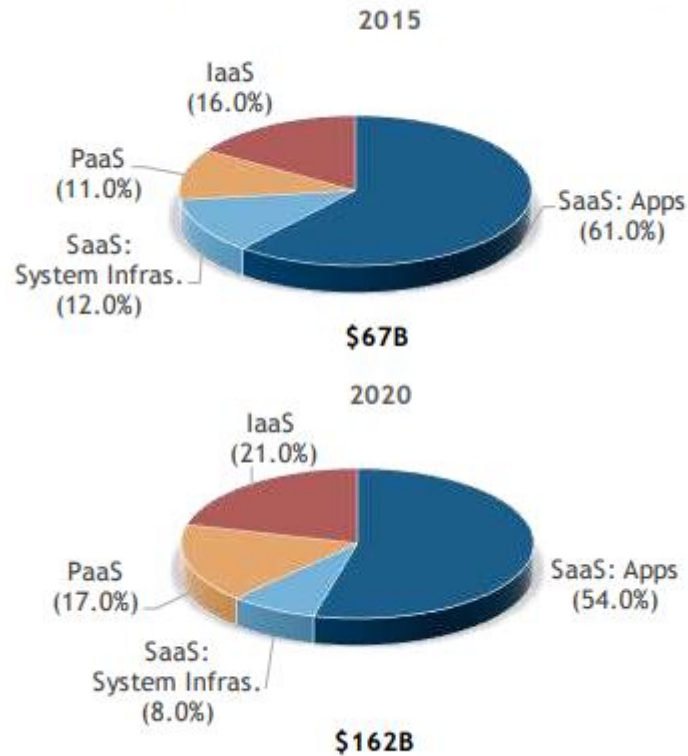
	Use of cloud computing	E-mail	Storage of files	Hosting the enterprise's database(s)	Office software	Financial or accounting software applications	CRM software applications	Computing power for enterprise's own software
	% enterprises	% enterprises using the cloud						
EU28	21	65	62	44	41	32	27	21
Belgium	28	57	66	47	47	47	33	28
Bulgaria	7	65	58	57	44	30	20	17
Czech Republic	18	76	49	31	40	31	22	21
Denmark	42	68	70	52	49	45	33	33
Germany	16	49	63	33	31	26	18	19
Estonia	23	64	53	25	45	53	17	25
Ireland	36	70	75	46	48	35	30	21
Greece	9	68	58	45	50	18	23	27
Spain	18	69	69	59	39	27	29	29
France	17	61	69	55	36	28	29	18
Croatia	23	76	56	47	50	45	15	23
Italy	22	85	41	39	35	19	19	10
Cyprus	15	72	72	33	52	26	29	18
Latvia	8	68	53	49	33	38	23	25
Lithuania	17	74	56	53	38	45	26	36
Luxembourg	19	64	69	51	47	30	28	24
Hungary	12	69	54	37	49	34	26	22
Malta	28	73	64	38	48	20	21	19
Netherlands	35	59	64	65	48	55	40	20
Austria	17	55	61	36	36	25	26	24
Poland	8	69	57	43	37	28	26	19
Portugal	18	76	62	37	45	32	26	34
Romania	7	68	52	51	37	43	19	27
Slovenia	22	61	49	37	48	33	23	22
Slovakia	18	82	48	35	50	45	18	23
Finland	57	74	63	45	51	46	34	14
Sweden	48	66	68	45	44	44	28	24
United Kingdom	35	62	69	41	49	35	31	24
Norway	40	70	71	64	54	53	39	31
FYR of Macedonia	7	71	56	54	47	39	22	25
Turkey	10	:	:	:	:	:	:	:

Turkey: data for subquestions not available.

2.3 Eurostat, Use of cloud computing services in enterprises, 2016

Additionally, according to the same whitepaper SaaS is and is predicted to remain the most used delivery model for cloud computing,

Worldwide Spending on Public Cloud Computing by Type (\$B)



Source: IDC, 2016

2.4 IDC: Worldwide Spending on Public Cloud Computing by type, 2016

2.3.1 SaaS Model Analysis

From a technical perspective, the cloud providers are responsible for maintaining and updating the tools they provide - deploying new features, releasing bug fixes etc.- and are mainly following a multi-tenant architecture; that means that all users are utilizing the same version of the software at any given time, contrary to how on premises software worked.

SaaS businesses usually follow a subscription based business model allowing the client to pay as they go based on their needs at any moment. As SaaS is only dependent

on cloud but not to any local configuration, any SaaS tool is accessible from any place, anywhere if the client has an Internet connection. Based on how a user is accessing an application, a thin client (browser), a dedicated client program (mobile app) or both can be utilized, based on the application's functionality. A user only needs to log in to their account to access the system and their data. If any configuration is required to meet an individual company's needs, then it is implemented as a feature/setting - it can be enabled or disabled by either the user directly through their account, or by the provider, but these are options allowed by the common codebase - without custom code snippets solely functional for a single client.

While on premise software is still used by companies, SaaS is taking over strong. Outdated, heavy monoliths of tools are slowly getting obsolete, as companies replace their software with lightweight yet robust applications to improve their workflow. While previously companies were looking for a single tool (usually heavy-duty ERP tools) to cover ground throughout all departments, SaaS allows businesses to own their software stack by integrating multiple tools with each other.

Besides SaaS transforming a company's software stack, the introduction of cloud computing is opening up the Internet of Things while promoting innovation. Today, you can find a SaaS tool for everything, and within categories the competition is forcing SaaS companies to push their limits and think of how to meet not only today's but tomorrow's needs as well. And taking a quick look at the new SaaS products that pop up every day, it is definitely noticeable that modern companies are shifting from workflow to automation, so they have more time to do the important things; instead of having to deal with an outdated system and input heavy processes, more and more AI and machine learning functionality is making business as usual more efficient.

2.3.2 Characteristics and Benefits

A list of common characteristics for SaaS tools easily showcases the benefits of using it compared to a legacy, on premise software.

1. **Quick setup and cheap customization:** For clients that use SaaS, setup can take just minutes for a user to sign up for an account, and any configuration for their own account can be easily made through the interface of the application, as it is offered as part of their features. There is no hidden cost to implement further customizations.
2. **No installation required:** As long as a user has Internet connection, they can access SaaS products immediately. This enforces collaboration and makes things easier, especially since companies are now looking for talent on a global scale and allow remote access, enabling employees to work from all over the world.
3. **Accelerated feature delivery and automated version upgrades:** SaaS companies, as we will see later, are product centric and usually follow agile methodologies and practices, with short development cycles and quick if not continuous delivery. Due to this, customers can get their hands on an improved service quicker, and get more functionality/features many times a year. All this, with no effort from the client.
4. **Scalability:** Traditionally, SaaS companies follow a pay-as-you-go business model which allows all types of businesses to get on board, and adjust the necessary resources they need from the tool (and pay accordingly) at any given moment. That allows small businesses to avoid hardware and expensive legacy software licenses, requiring a substantial upfront investment, only spend money on what they actively use, and adjust the necessary resources in minutes by

upgrading or downgrading. As a company grows, the offerings of the tool can grow with them in the click of a button.

5. Increased security: While security in the cloud is a rather controversial topic, mature SaaS products usually are more secure than on premise, traditional software. Since the data are distributed in multiple server locations, they are in more secure hands than being in a single office. All data is encrypted by the provider, companies make an effort to get certified for ISO 27001, the Information Security Management Standard to ensure security, and usually also run a bounty program to eliminate security bugs from being in the application.
6. Easy integration via APIs: Application Programming Interfaces (APIs) are particular sets of rules that software programs can follow to communicate with each other programmatically. Multiple tools offer Application Programming Interfaces (APIs) and integration protocols, allowing companies to have access to their raw data as well as create integrations for internal company use.

All indications point to SaaS being the future of enterprise software. But a delivery model alone cannot make you a successful software; In the following chapter we will be looking into available project management methodologies, which of those can be best utilized for a SaaS tool, and how it assists in its development.

SaaS highlight - Workable

Workable is a technology startup building modern recruitment software for applicant tracking with more than 6000 clients in 80+ countries - and at the same time, one of the most successful startups in Greece, and was originally created as an alternative to archaic, clunky, out of date hiring software the founders had come to use in previous jobs. Starting off with \$500k of seed investment back in 2013, it managed to receive \$27M in their Series B back in September 2015 and are going strong to become the world's best applicant tracking system. The Workable team creates beautifully designed, simple recruiting software for fast growing companies, while making it easy to collaborate and own your recruitment.

With a fast growth of its own, the company has had an employee growth of over 40% in the last year and continuously hires. With offices in Athens, London and Boston, it works towards streamlining the hiring process and empowers teams.

In a very competitive market, recently joined by Google's own applicant tracking system, Google Hire and Microsoft's Dynamics365 Talent tool, it continues to grow to help companies hire better.

3. Managing SaaS

While project management is a discipline whose methods could apply to any type of project, this is not always the case. It is not of course the case that the core principles of project management are not true for all types of projects and businesses - but since there are differentiators depending on the case, multiple branches for project management methodologies have arisen to fit each need. In this chapter we will be examining the traditional core principles according to the Project Management Institute (PMI), as well as multiple methodologies leading to the one we consider best fit for SaaS. After this, we will be delving in a macro view of the development of SaaS to see the results of multiple project management life cycles and how they can be defined in an early stage.

3.1 Project Management according to PMI

According to the Project Management Institute (PMI), the nonprofit professional organization for project management based in US, "project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements". The concept of a project has a broad scope for project management, which is the main reason why there are multiple Project Management Methodologies (PMM); to customise the techniques and best practices used with the goal to optimize the management of each separate project type based on specific characteristics. But regardless of which methodology we're focusing on, the project management activities usually refer to the following major process groups:

1. Initiation
2. Planning

3. Executing
4. Monitoring and controlling
5. Closing

The method a project is going to be run by is up to the project manager to analyze and select per case, as it highly depends on the project characteristics. In the IT world, the projects usually have unique characteristics so the set of skills, methodologies and techniques need to support those characteristics and successfully dodge possible risks.

3.2 Project Management Methodologies

What the PMI describes in detail in the Project Management Body of Knowledge (PMBOK) is, as we mentioned, a set of standards, key concepts, emerging trends and details on how one could apply tools and techniques to a project. In the context of a project, the project manager can decide to use one or more Project Management Methodologies to successfully implement what is outlined in the standard.

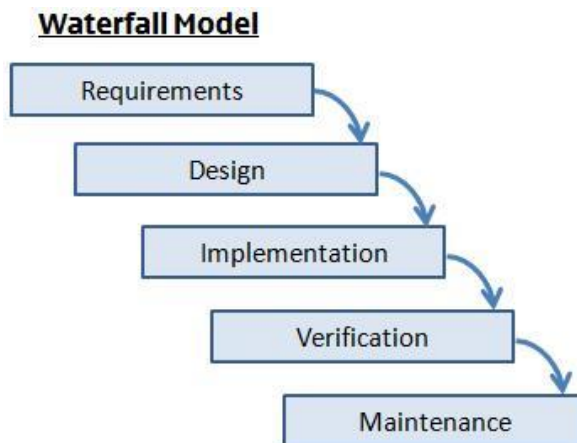
In the next paragraphs we will be looking into the most popular methodologies, and see which one is ideal for software development.

3.2.1. Traditional Project Management with Waterfall

One of the most prominent methodologies, and widely used for years is the Waterfall model, born in manufacturing industries. As it dictates a sequential flow from one step of the project to another, changes to the original plan are not welcome as they are very costly. The level of structure demands that all requirements are available in the

beginning, and that nothing will change until the project finishes. Waterfall consisting of the following phases:

1. Requirements Analysis
2. Design
3. Implementation
4. Testing
5. Maintenance



As each step must follow the previous one, flexibility is minimal and delays an often-seen issue, as one step needs to finish for the next one to start. While widely used, this is not an ideal methodology to be utilized in software development in today's fast paced environment - at the time it was adopted for software development, no other models were available to be adopted.

3.2.2 Agile

Contrary to Waterfall's life cycle, the agile project management methodology opens up an iterative/incremental approach in managing software development projects, focusing on continuous releases and uninterrupted communication with the customer to get feedback and continue development. The methodology emerged in the form of the Agile Manifesto, created in February 2001 by seventeen people working on software development (Agilemanifesto.org, 2001).

In the past few years agile has continuously gained recognition as the best fit for managing software projects, and a root cause is definitely the increased complexity of software development in the modern era. With cloud computing scaling from barely there to an enterprise tool, uncertainty about the requirements of the market is increasing and any team needs to be able to respond to change fast to catch up with both the client and the competition.

The concept of Agile stems from Toyota's lean manufacturing, with the ultimate goal of optimizing processes in a way that eliminates waste and addresses changing customer needs, while retaining transparency. Opposite to waterfall's buildup for a single "Big Bang" of a release, agile allows teams to break down their work into significantly smaller chunks, making it easier for them to respond and adapt to changes in requirements - a common occurrence in the software sector, as the uncertainty of what the client wants is a basic characteristic of the market.

With disruptive technologies such as the transition from on-premise software updated maybe once a year and manually maintained to cloud computing, agile is leveraging project management to allow teams to respond in high uncertainty environments instead of waiting for the total of requirements upfront; it allows development to become an exploratory process, much like life itself. Teams are able to explore in short cycles, get feedback and improve on themselves and their processes.

The principles described on the Agile Manifesto can be summarized as follows:

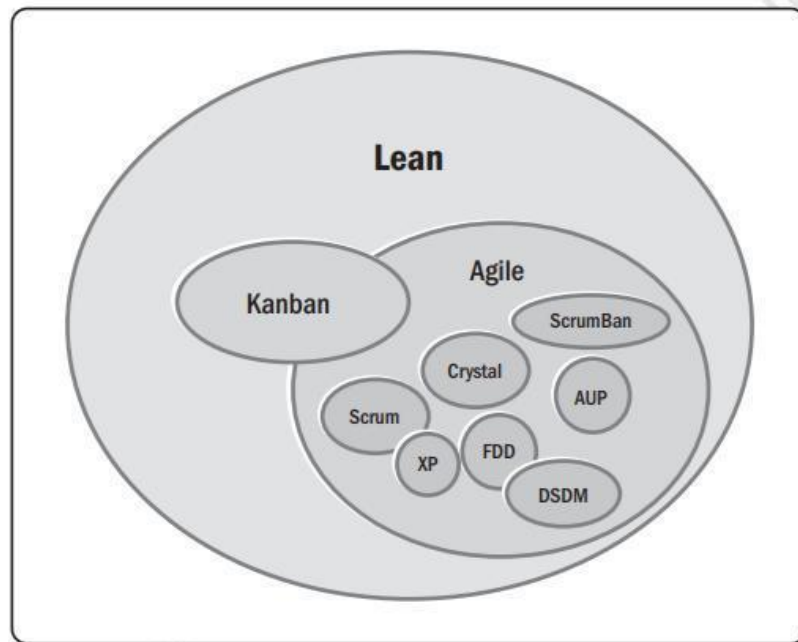
1. Focus on the customer: Client is on highest priority, and is collaborating with the team instead of negotiating about the project.
2. Team delivers working software frequently in short bursts (iterations).

3. Business and Engineering work together daily, and along with the sponsor they should be able to maintain a constant pace indefinitely.
4. Working software is the primary measure of progress
5. Instead of “obeying” under a project manager, teams are self-organizing
6. Face to face communication is most efficient.
7. Changing requirements are welcome within an iteration.
8. Projects must be built around motivated individuals that are provided the necessary environment and support.
9. Attention to technical excellence and good design enhances agility.
10. Simplicity is essential.
11. The team needs to reflect together and discuss how they can improve the way they work in regular intervals.

In summary, agile demands that in each iteration the requirements will be discussed and agreed upon, a small part of the software will be built and tested within the iteration, and in the end of the iteration the software’s future along with new requirements is discussed upon, offering the team the flexibility to adapt and deliver value often while the product evolves. Clearly, this requires a shift from a project mindset that looks at the whole picture once in the end, to a product approach, where teams have a single goal to create/improve a product and have the opportunity to discuss about it often.

3.3 Scrum

As we saw previously, agile has some core principles that assist in managing software development - but agile is not a single thing. It is an umbrella term for multiple iterative/incremental methodologies that can be utilized for software.



3.1 Agile is a blanket term, Agile Practice Guide by PMI, 2017

Scrum, the most widely used subset of Agile, is a lightweight process framework for development, and offers a set of best practices a team must be consistent with.

Scrum was formalized as a framework in 1995, and was developed by Jeff Sutherland and Ken Schwaber. As explained in detail in Sutherland's book "The art of Doing Twice the Work in Half the Time", Scrum is designed for developing and sustaining high complexity products, and while its success dictates following the core principles as described, it allows a team to also employ various processes and techniques.

The framework is designed for teams to use for iterative development, so they can break their work into actions to be completed in sprints - iterations of (usually) two weeks - track progress, collaborate efficiently to reach a common goal, and finally plan for the next sprint. The common goal is to deliver working software at the end of each sprint. Scrum is composed by specific roles, artifacts and events, as well as rules these components must abide by.

The *basic roles* are:

- The development team: a cross-functional, self-organizing team of three to nine members, equipped with all skills necessary to deliver working software at the end of each sprint.
- A product owner: the person that defines the requirements for implementation, and is responsible for maximizing the value of the product. The owner represents the stakeholders as well as the voice of the customer.
- A scrum master: responsible for making sure Scrum principles are applied properly, and preventing distractions from reaching the team; the servant leader of the team.

Artifacts are concepts utilized by the team to reach their goal:

- Product Backlog: Short groups of requirements described in detail in forms of stories for the developers to implement. It is maintained, prioritized and updated by the product owner, and captures the set of requirements for the product to be complete. Stories represent the work for the development team, and are evaluated in terms of a effort per story by a ranking system using story points, usually

following the Fibonacci algorithm. The estimates assist in giving a sense of timeline for delivery, as well as in filling up the Iteration backlog.

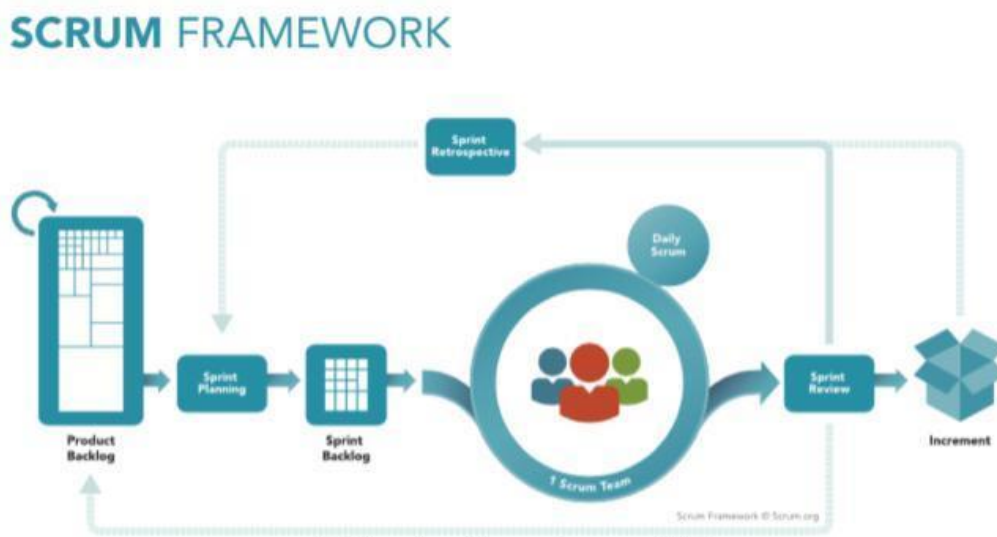
- Iteration (Sprint) Backlog: The sprint backlog lists all the stories the team has committed to deliver within the next sprint. It is filled by the first stories of the product backlog (as they are prioritized) according to the estimations the team has provided - the total of points delivered per sprint on average is a metric typically used to provide a grasp on what each team can deliver per sprint in terms of effort.
- Increment: Product increment is the total of what the team delivers in the end of the sprint, along with everything else that has been delivered before that. What is delivered in the end of the sprint should be the completed work for all stories added in the sprint, in the form of working software ready to be released.

To successfully practice scrum, some *events* must religiously be practiced by the team to ensure coordination and successful collaboration:

- Sprint: a time box of usually two weeks, but customized depending on the team's needs, in which the team needs to implement, test and deliver specific requirements
- Sprint Planning: A recurring meeting where stories from the top of the backlog are discussed with the team in detailed, evaluated and the team agrees on what can be delivered in the coming sprint.
- Daily scrum: a daily, short (up to 15 minutes) meeting where each member of the team replies to 3 questions: what they did the previous day, what they will do the coming day, and whether anything is blocking their work.

- Sprint review: At the end of each sprint, the team meets and demonstrates the implementation to the product owner and the rest of the team and reviews what was not completed.
- Sprint retrospective: This event also usually takes place at the end of the sprint, and the team shares what went well during the sprint and what didn't. This is the chance to discuss best practices, and what the team can change to improve collaboration and effectiveness.
- Backlog refinement: While this event is not necessarily strictly present in every sprint, it takes place when the team needs to discuss and refine stories that have been prioritized in the backlog. It is reviewing the stories by the team, and making sure that everything is clear and executable when each story enters a sprint.

Later on, we will see where in creating a SaaS from start Scrum fits and how it helps manage engineering teams and product development; but it's important to remember that scrum is not a framework solely applied to software development. Enterprise scrum has been created to bring a more holistic framework to a company that defies departments, where it extends the use of Scrum across the organization.



3.2. Overview of Scrum, Scrum.org, 2017

Spotify, a paradigm company customizing Agile to their needs

Spotify is a SaaS company created in 2006, with current revenue of \$2.18B, more than 1,600 employees in 20 different locations, and is predicted to end 2017 with 70M paying users and \$5BN in revenue. The team started off with a handful of people and a monolith of an application to manage and improve using Scrum, but with growth it was evident that they needed to experiment to scale successfully. Here are the actions that helped them become one of the most innovative SaaS companies today:

- Scrum made optional instead of “Scrum is all” for whoever it works: agile is more important, bringing the focus on principles instead of practices.*
- Revamping the organization: Squads, chapters, tribes and guilds were introduced. Squads are autonomous product teams responsible end to end for the delivery and maintenance of parts of the software, enabled to make their own decisions. Chapters are groups of people with the same skillset. Tribes are the work environment, while guilds are people not necessarily specialized in a topic but that are interested in learning about it.*
- Less standardization, more cross pollination. No real rules about what each team should use or which process/tool one should utilize, but if something works it will be shared with teams around.*
- Test automation and continuous delivery: With a handful of developers familiar with the application it was easy to make changes. Now, since there are so many independent product teams within the company, the architecture of the application has completely been redesigned to support ownership of code for every team as well as independent, often releases, as well as robust regression suits and unit tests per team to support continuous delivery.*
- Culture. Open office culture, internal open source model and culture in engineering to spread knowledge and a “fail fast, learn fast” principle in place, the culture has grown along with the company and allows increase in efficiency and disconnects politics from frameworks. Teams trust each other, and other users trust that each team will be responsible and will efficiently release their own code in production successfully.*

Despite all these changes, Spotify believes there’s still a lot to learn; that’s why it encourages experimentation in every way and motivates each individual to innovate from small to big.

4. Maturity of a SaaS business

After having a glimpse of what a SaaS business is, and how it is managed on a day-to-day - or iteration to iteration- manner, it is time to zoom out and take another, more macro-oriented view of it.

Starting from scratch, a SaaS business goes through multiple stages to reach not only profitability but also success and establishment at its sector - those stages can be mapped into different maturity levels a company inevitably goes through as it grows.

According to the bibliography, there are multiple maturity models for businesses based on multiple criteria - with perhaps the most famous of them all being the Capability Maturity Model (CMM) specifically for software. CMM was originally created in the 1990s by the Software Engineering Institute to advance software engineering methodologies, and assesses an organization against a scale of five process maturity levels. Each level ranks the organization according to its standardization of processes in the subject area being assessed.

Since CMM is a process maturity model, it is not what we will be going into, as we look at a SaaS business as a whole. While a process centric thinking is definitely important, it cannot define IT business maturity out of the box. We will be looking at two different maturity models according to the bibliography, and refining each stage based on our experience to generate a “custom” maturity model to be used in our next steps of mapping SaaS processes.

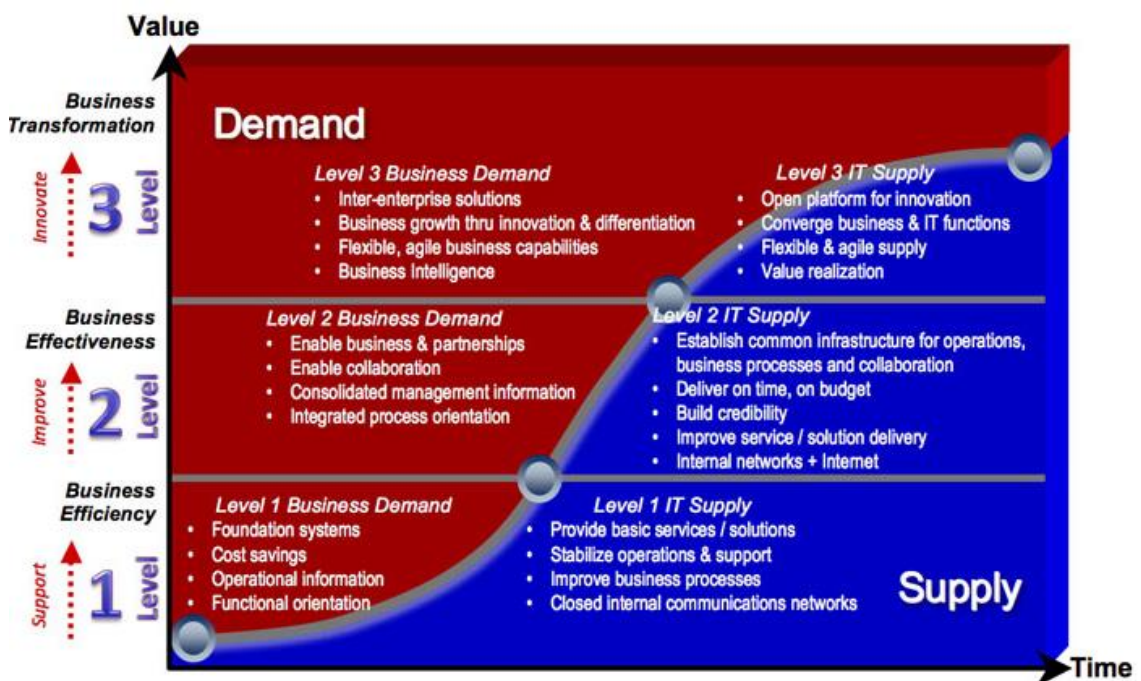
4.1 The Business - IT Maturity Model

Originated back in the 1990s by Vaughan Merlyn and a team of researchers in E&Y, the Business - IT Maturity Model (BIMM) was such a breakthrough from the

widely spread CMMI because it combined the maturity stages of both the supply for IT from a company's point of view, as well as the demand for it directly from the market. While combining these two sides of the coin, the model brings to the surface interdependencies and allows organizations to see value in the long term.

In its basic form, BIMM is a S curve having the market demand on the left side, and the corresponding goals of IT supply on the right side. The enumeration of levels is usually arbitrary, but the three levels depicted in the figure showcase its simplicity for most cases.

Much like in most markets, the maturity curve is cumulative; just because the latest iPhone has a facial recognition feature, it doesn't mean that the need to make phone calls and send text messages isn't there. On the contrary - the "feature" aspect of each previous maturity level becomes the norm, the bare necessity for the next one.



4.1 Business - IT Maturity Model, Vaughan Merlyn, 1990s

At **Level 1**, business demand is looking for efficiency - there are processes and steps and actions that need to be taken and currently get done messily with dispersed

information from multiple tools, take too long and are expensive. To this demand, IT responds with automating those processes. Basic solutions are provided, usually to specific business units, and give time to the people to optimize the processes, stabilize what they offer and connect tools to make finding information easier.

For **Level 2**, business sets a goal to be more effective than efficient. While working in separate business units before, now is increasingly approaching problems with cross functional teams to complete end to end processes. Supply focuses on supporting current processes and stabilizing enterprise systems (ERPs, CRMs) and allow of a more effective use of the data on hand with Business Analytics. It also establishes a common infrastructure, inviting opportunities for selective partners to join in.

Finally, in **Level 3** the demand is nothing like what it used to be. The search is no longer up for a good automation tool, but for a complete, flexible solution that will become a partner to the company and will help it grow through innovation. At this level, IT is now focusing on being agile and flexible to meet constantly changing needs and create innovative offerings that will become an organization's standout factor. With a strong focus on strategy and adding value to their offerings, IT is taking advantage of the best technology has to offer - from tools to AI.

The model is a management tool, purposefully simple to enable different types of stakeholders - like business and marketing people and developers - to participate in a discussion about where an organization is and how it could reach the next level of maturity. By analyzing the equilibrium between demand and supply, stakeholders can align their priorities on resource allocation, strategy of the business and accordingly, strategic use of their technology and people, and discover more ways to increase the value of their offering. Systematically using BIMM in an organization assists in detecting gaps between supply and demand, helps in investigating how the model phase can affect and

improve a specific business unit and can give a clear picture of where a company is at any given moment.

An important thing to point out here is that moving from one level to another does not happen from one day to the next just because a company has “successfully” completed the first one. Moving up towards a higher maturity level does suggest that what the company did so far, it did well, but the transition from Level 1 to Level 2 calls for change in mindset, strategy, tools and techniques, and moving to Level 3 requires a breakthrough for innovation and becoming a partner, not just another service to your end user.

4.2 IT Infrastructure and Operations Maturity Model

In October 2007, Gartner introduced their IT Infrastructure and Operations Maturity Model that helps leaders examine the maturity of their organizations in four different aspects: *people, process, technology and business management*. As a tool, it assists organizations in detecting where in the scale of maturity they are at any moment, and establish a roadmap for higher maturity levels for partnering with the business and optimizing the offered services.

While previously we saw how BIMM can assist in offering a complete outlook of the stages of a business combining demand and supply, Gartner’s I&O reminds us that as technology evolves and is even more ingrained throughout an organization, all supporting sections of it must evolve to support it. People, processes and technology are prerequisites for process improvement, and they are all driven or constrained by the way they are managed.

The model evaluates these four dimensions as the process, people and technology being supported by business management, and together they generate business value assessed by the following metrics:

- Economics
- Quality of Service
- Agility
- Customer Satisfaction
- Business Contribution

There are six maturity levels:

- Level 0: Survival - Bare minimum focus on Infrastructure and Operations
- Level 1: Awareness - Realization that I&O are vital to the organization. Small actions are taken to obtain visibility and control.
- Level 2: Committed - A more organized environment for IT support, and increased customer satisfaction.
- Level 3: Proactive - Creation of policies, standardized processes and quality control for better change management and a more efficient environment.
- Level 4: Service Aligned - Customer focused, proven and competitive IT service.
- Level 5: Business Partnership - Trusted partner to the business for increasing value and competitiveness.

In the next chapter, we will be mainly focusing on the BIMM model while taking advantage of elements from the Gartner I&O maturity model for processes relevant to Business as Usual (BAU) to create a custom maturity journey for a SaaS service and predict its path towards maturity via process mapping.

Maturity and Key Performance Indicators

When making an effort to efficiently track the maturity of your organization, what is counted is what matters; if you don't track something, it won't change or bring any issues to the surface. As it would be practically impossible to track every possible Key Performance Indicator (KPI) from the moment a business is born until its exit or doom, short and long-term goals should be explicit at any point and be acted upon; counted, tracked, revisited methodically. What's important to mark is that this should not disallow or make sudden changes to strategy hard or impossible to navigate; along with the actual development, tracking should be methodically set up in cycles, predominantly in an agile environment to ensure the company tracks what's important for it and its future at any moment. Besides that, it can serve as the single source of truth about which maturity level the company is at now, provided that the company has set up long term goals based on a maturity model.

Tracking KPIs is necessary for multiple stakeholders to enable decision making in multiple forms. For example, Operations need data on how each operation goes to make Business As Usual (BAU) more efficient, or make corrective actions. In a similar fashion, product needs to be aware of how the latest release is performing, or if a low-usage feature is having a negative effect on the product, or if any bugs when live on production and were missed in testing; this could lead to deciding the next big feature, cutting off a legacy feature that brings performance issues and little value, and improves automated and manual testing. Of course, these are smaller decisions; financial data are usually the biggest part of KPIs tracked, and assist in making significantly bigger decisions and help track on whether the strategy implemented works as planned.

5. Case Study

In this chapter we will be following a SaaS company starting from scratch, its journey through different maturity stages following a custom maturity model, and its processes are affected from stage to stage. We will be discussing about a tool in the Human Resources Tech category just being created, that allows the HR department of a company to onboard new employees, manage payroll and track time off, as well as offer a database for employer data and records. We will go into detail about each maturity step for the company from even before it was released, and utilize our experience and process identification as well as rapid process discovery for successful startups to identify and morph processes for each step. While it would be impossible to record all processes for each maturity level, we will be analyzing only part of the processes run in a company to showcase the differences between maturity levels.

5.1 Tools

For the analysis of the case study we will be utilizing Business Process Model and Notation to map out some processes for each maturity stage of the business, and we will be using the free tool yEd to implement the diagrams. yEd is a graph editor desktop app, offering options for multiple modelling languages including BPMN.

5.2 Maturity Model and Analysis

Earlier we saw in depth two different maturity models, BIMM and Gartner's I&O, and both could be applicable for tracking a business' maturity - but for the context of the

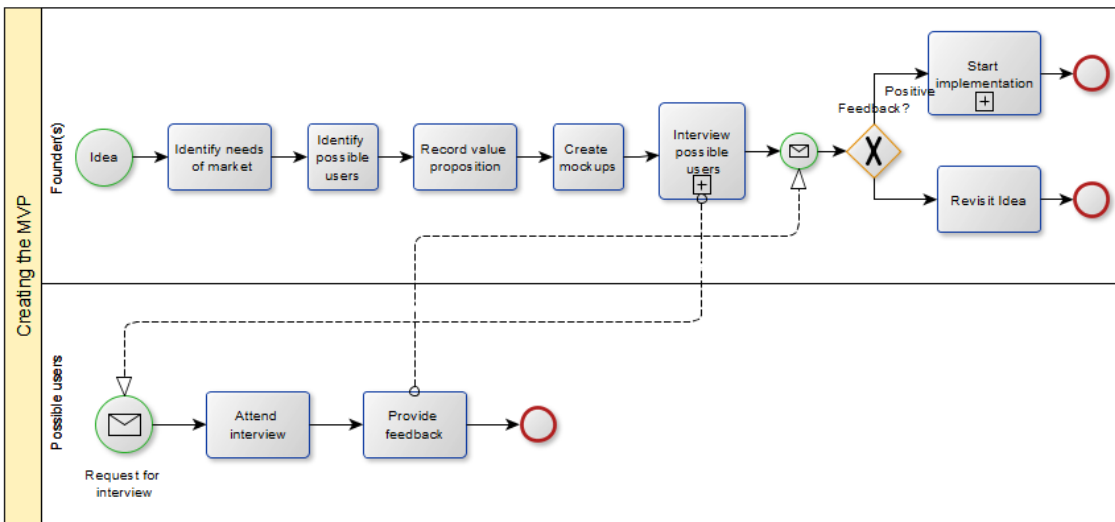
SaaS tool we will be delving into, we will be utilizing elements from both maturity stages to represent an example growth trail for a SaaS company according based on experience as well as from tracking the transition of other SaaS tools in similar phases.

Our suggested maturity model that is utilized for process mapping in this thesis is comprised of the following levels:

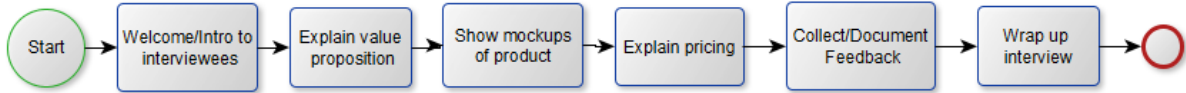
Level 0: Creating the Minimum Viable Product

This is the start of the tool, where it has not yet been created. The goal of this step is to investigate the idea behind the product to be created, verify that there is a need for it and implement a minimum viable product (MVP) to release to the market as soon as possible.

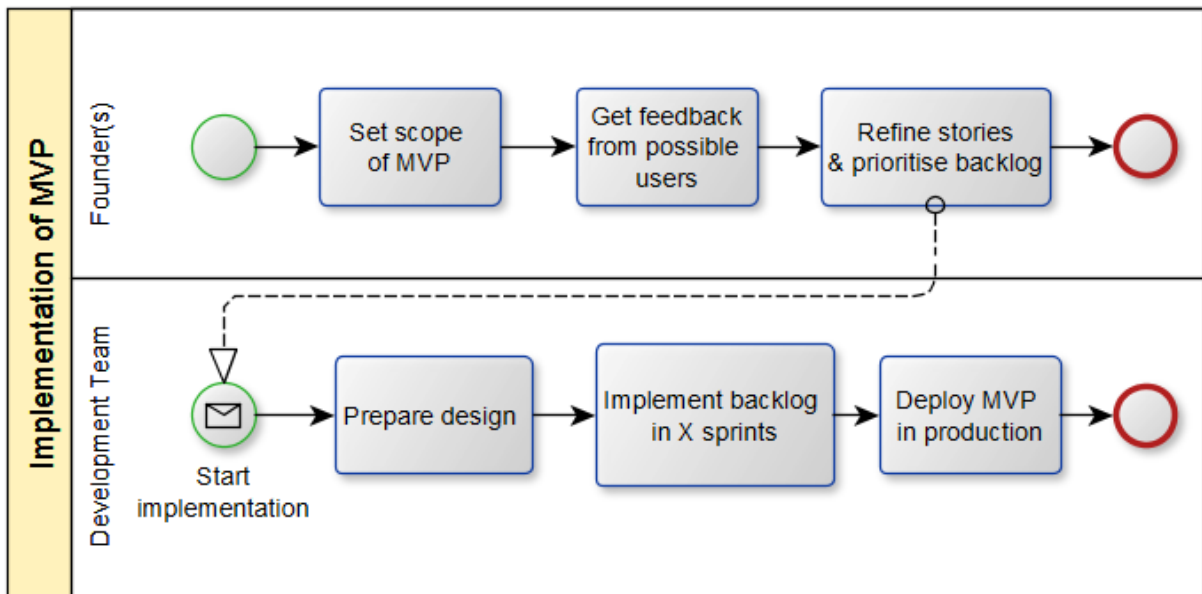
Taking a closer pool in the BPMN diagrams below, we can see everything starts from an idea - investigation is required on whether the idea represents a part of the market, and preparation is required from the founder so the hypothesis can be validated through interviews with possible users and customers. Naturally, this is a high-level representation for it, as we do not delve into funding details, setting up paperwork for a company establishment etc.



5.1 Creating the minimum viable product



5.2 Sub-process of MVP - Interview possible users

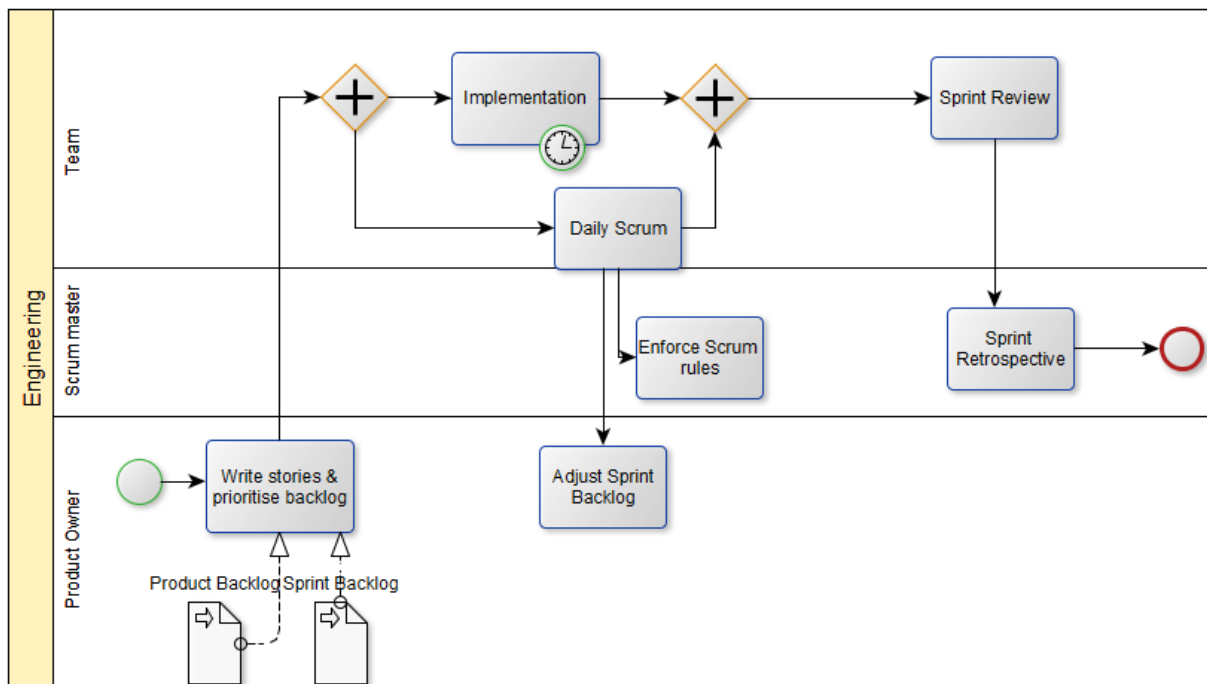


5.3 Sub-process - Implementation

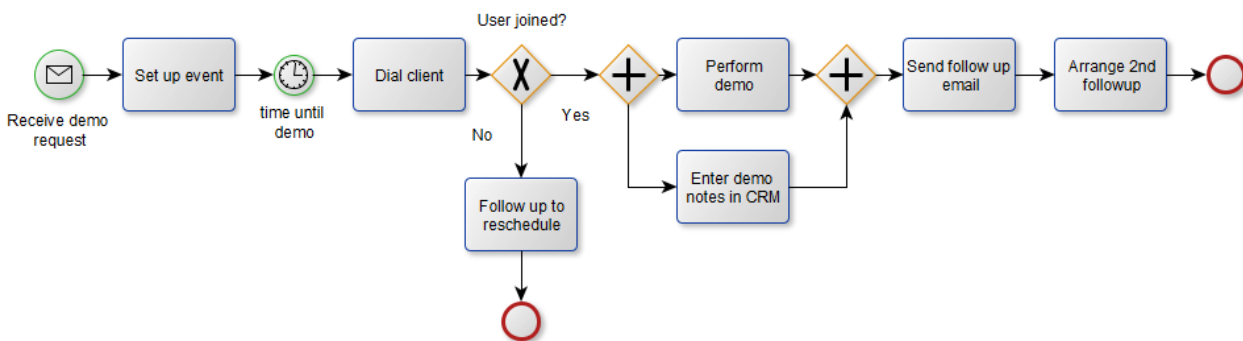
Level 1: Support - Getting out there

In this level, the tool is live and has gathered a small client base. There are few actual processes, and most things are ad hoc decisions made on the spot, like handling bugs where support can just speak to the developer next to them about a bug reported and ad hoc plan for a fix without a ticket. The demand is for a lightweight tool that assists in daily operations and takes some pain out of them, and does not need to be bloated with many features. It is to serve specific functional requirements of the HR department only, not to replace existing processes/tools. This is what the tool would respond with in their offering, building a simple, easy to digest solution that is on its way to meet this standard feature release after feature release. The team is still small, and only some advertising is made, while the tool has low cost and offers a free trial. Implementation of features is responsive based on the market, and the rest of the departments in the company are also responsive based on day by day operations.

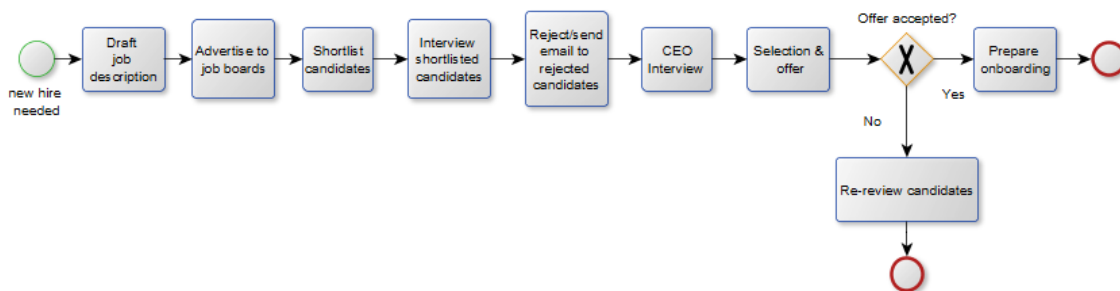
In the diagrams shown below, we get insight on how engineering works - following the scrum principles as explained earlier - and how sales are handled at this level. The tool is offering a free trial of the software, and along with marketing and search engine optimization signups and leads are generated, and if a detailed demonstration is requested from the user's side, it will be responded. Lastly, we see how the team hires - this is a straightforward process with few means utilized, as the team grows slowly and has not yet focused on setting up an employer brand or increasing recruiting efforts. There is no official approval flow for a job opening, and the evaluation stages are few and direct.



5.4 Engineering process



5.5 Sales team handling a demo request



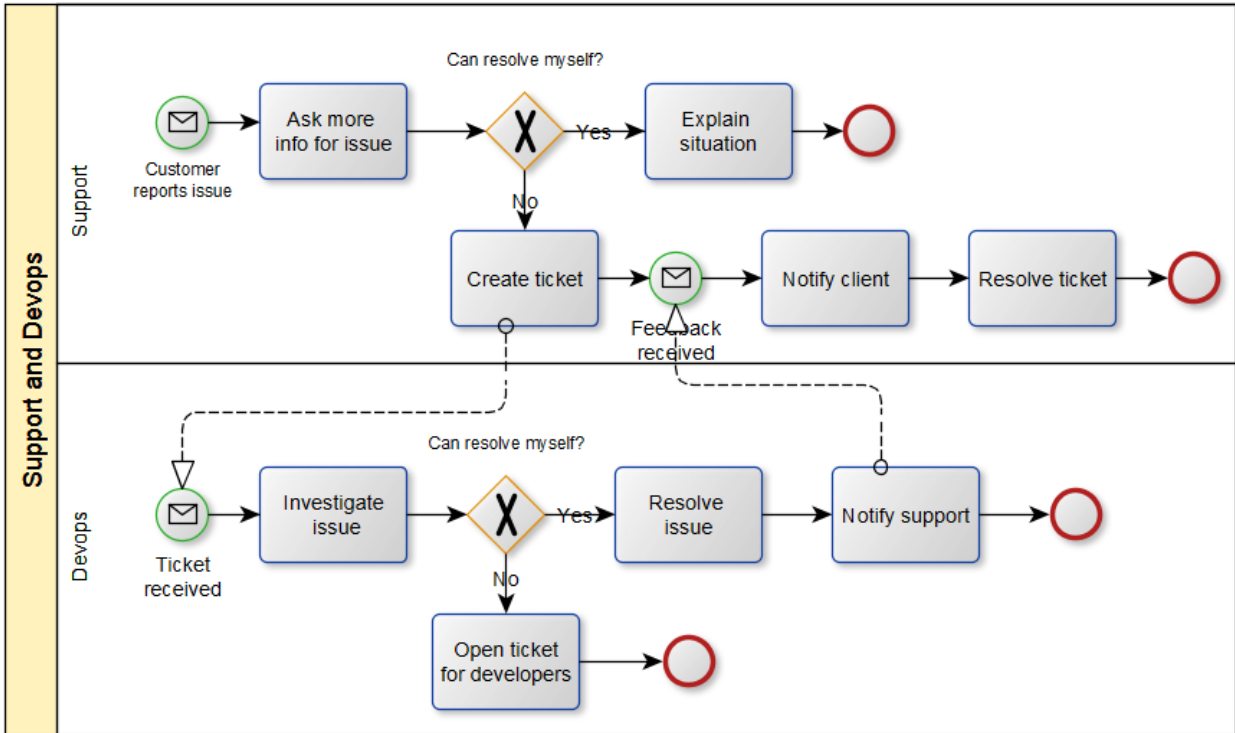
5.6 Hiring Process

Level 2: Awareness

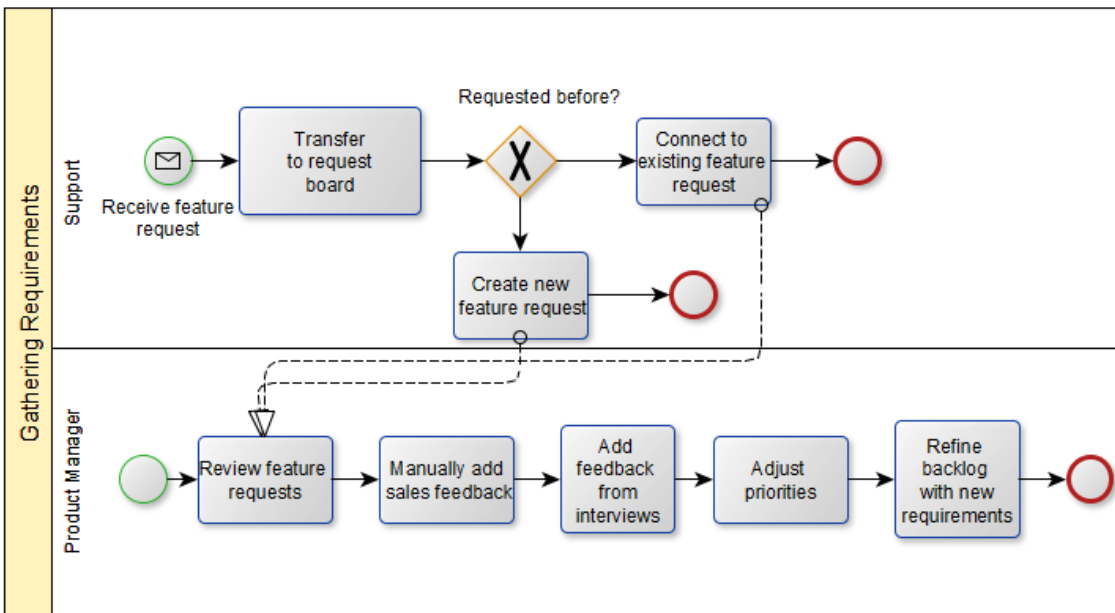
At this level, business is starting to opt for efficiency. The client base is now bigger, and there is steady growth that needs to be made sustainable, so the business puts effort in gaining visibility into the data and starts documenting some structured processes. While before operations only worked as reactive, now there are ad hoc cases of proactive thinking. The business still targets small and medium sized businesses. Some processes are automated and tools are integrated to optimize data analysis. Customers are using the product, and the goal of the business is to increase happiness and engagement.

Processes start getting in place. What could previously have been a short walk to a colleague's office for a question in regard to a bug now has official steps to be taken, as shown in the diagram 5.7 below. There is a separate department for technical support (development and operations - DevOps) and has a process to communicate with support when needed.

Gathering requirements is now a bit more complicated, as there is feedback from multiple customer facing teams to be reviewed - this is done manually and is repeatable, but is time consuming and requires actions from multiple employees that can easily be forgotten. This feedback will be used to shape future features to be implemented.

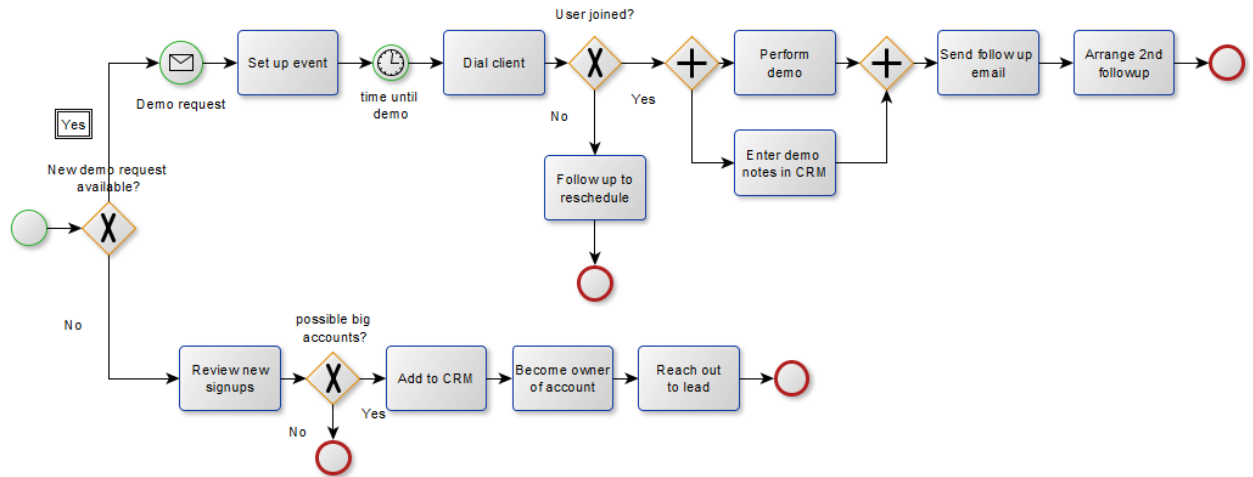


5.7 Support and DevOps resolving a customer issue



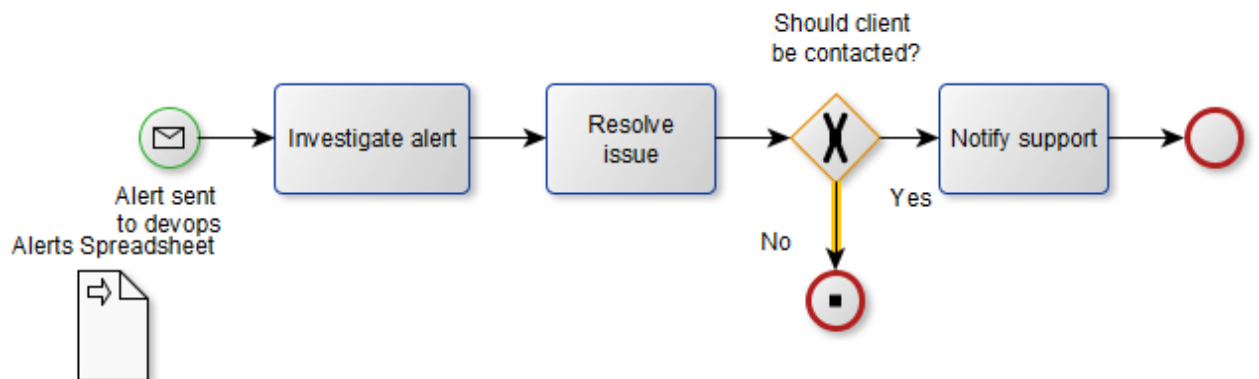
5.8 Gathering requirements and customer feedback

As the company has grown, sales is also trying to be a bit more proactive and now incorporates a trial list review to proactively reach out to customers, with a goal to build a relationship and convert the trial to a paying plan.



5.9 Sales with proactive process

Due to a more proactive approach, the DevOps team also sets up a spreadsheet of usual issues users can face, and set up alerts so they can proactively resolve them without having to wait for the user to contact them, or worse, churn.

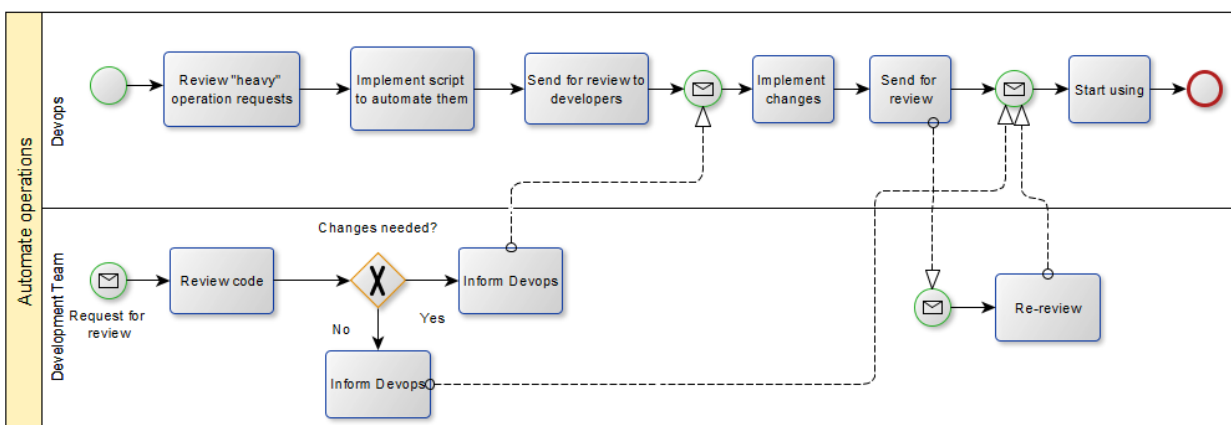


5.10 Alert setup for common issues

Level 3: Committed

A likely path for the business is to move upmarket towards targeting enterprise, and strengthen sales. As enterprise requires additional features than SMBs, the roadmap is adapted accordingly. Proactive behavior is now clearly showing up, as new departments for account management and customer success have emerged. Documentation and processes are standardized in most teams, and Business Analytics allows the business to improve the service quality. Processes for data security are also set up and standardized. Finally, the business invites partners for integrations to enrich their offering.

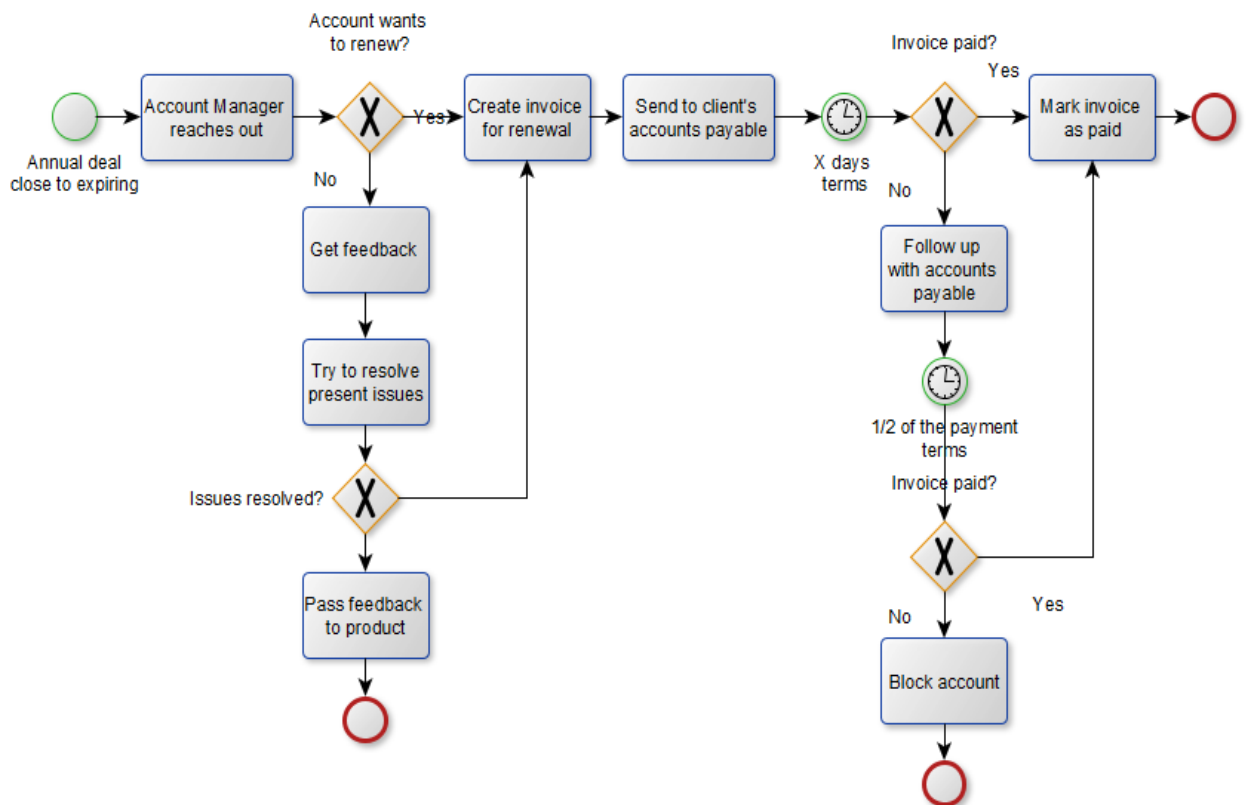
As the company grows, heavier operations need to be implemented such as deployments. As this takes up significant time, the ideal way to face it is to automate it to drop the time waste. As the DevOps and technical support teams will be writing scripts to automate said operations, and will be working on the new code, someone from the team will need to review the scripts to ensure they will work as desired.



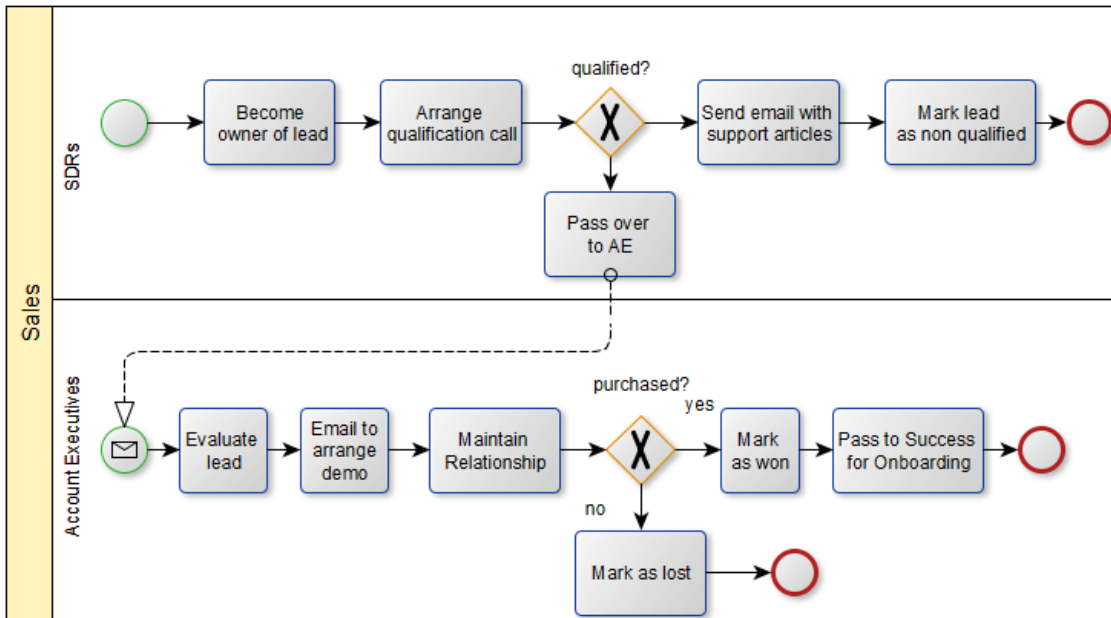
5.11 Automating time-consuming operations

Along with bigger plans for enterprise, a process is set to minimize churn. A new department emerges for Account Management, responsible for assisting users with any issues as well as ensure the account will renew their plan.

For the same root cause, the sales environment has radically changed. There are separate departments within sales: Sales Development Representatives (SDRs) handling small plans and evaluating whether an account will be a big or small plan, and Account Executives (AEs) that take over potential big plans qualified by the SDR team.



5.12 Account Management handles a plan renewal

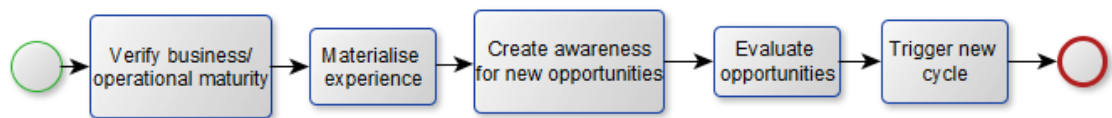


5.13 Sales environment is now composed of AEs and SDRs

Level 4: Business Partnership

The business is now efficient and effective, and is a competitive tool in its market for any type of client. It now focuses not on automation of clerical tasks, but on continuous innovation and incorporating business intelligence to the tool. It is a trusted partner to its clients, and strives to increase the value of its offering. In this level, processes are set and automated, but the culture is most important for the organization to retain its success. Promoting innovation within the company can certainly be promoted by processes, but the culture about being open, optimizing resources and thinking one step further than competition and even the client himself is the most vital part of retaining the maturity level and success. While optimizing everything existing, the company branches out and starts discovering new areas to delve in; this could refer to new product, new customer markets, new conclusions from experience that can create something new and innovative

- from process to product. Looking at the big picture, the process that can be depicted in this maturity step with accuracy is a metaphor for rebirth - and will continue by being recursive, as the company will move on to start something from scratch and create awareness for new opportunities, while retaining the gained experience. As time goes by, a cycle will end and a new one will begin, taking the organization to new grounds.



5.14 Maturity level 4, Essence of Business Partnership

6. Conclusion

In this thesis we introduced the simple principles of processes, building up knowledge to explain the philosophy of Business Process Management and its lifecycle. Through this analysis, we identified the three first phases - Identification, Discovery and Analysis - that later assisted us in our empirical analysis, identifying processes in an organization and putting them on paper along with its benefits.

Later we took a trip down memory lane on how on-premise software was the leader of software tools, and proceeded to introduce the Software as a Service era, analyzing the business model, its characteristics as well as benefits compared to the on-premise model. Following this, in an effort to understand how a SaaS business could be managed efficiently we dived in the core principles of Project management, and looked into different methodologies to detect which one can best assist such a type of organization grow to its utmost potential. After analyzing both Waterfall and Agile methodologies, we explained how Scrum as part of Agile can best be performed on a SaaS business and allow it to not only sustain the business but grow it as well.

As part of analyzing what SaaS is and how process mapping and different methodologies can help grow it, we introduced the concept of maturity of a business. Looking through the bibliography, we detected two maturity models that could be a fit for depicting the macro-view of a SaaS business life, and ended up combining them and joining them with personal experience to create a model more appropriate to SaaS needs.

We were able to specify 5 different maturity levels that can represent a SaaS journey from birth to maturity, and were able to present specific functions for each maturity stage and depict how each one is affected through this timeline. Through this,

we were able to model these processes using Business Process Model and Notation, pairing them with Swimlane modeling to showcase multiple departments and the flow of information.

We were able to achieve the creation of a concise model, progressively more detailed, with the end goal of depicting business transformation in the SaaS sector. Through this model, we also illustrated the concept of a modeling maturity mechanism and its analysis, fully proving that the cycle of business growth is recursive and can never be truly finalized. Mixing project management methodologies with BPMN, it became evident that the evolution of SaaS in truth is of regenerating nature.

Closing, this model is not necessarily confined to the virtual walls of SaaS growth - much like Agile and Scrum for engineering and other departments, it can be applied to other domains, and hopefully assist in the discovery of disruptive conclusions in the industry.

References

- Akram, A. (2017). *BPM and Software as a Service*. [online] Bpminstitute.org. Available at: <http://www.bpminstitute.org/resources/articles/bpm-and-software-service> [Accessed 25 July 2017].
- Blank, S. (2013). *The four steps to the epiphany*. [Pescadero]: [K & S Ranch].
- Boillat, T. and Legner, C. (2013). From On-Premise Software to Cloud Services: The Impact of Cloud Computing on Enterprise Software Vendors' Business Models. *Journal of theoretical and applied electronic commerce research*, 8(3).
- Bpmn.org. (2017). BPMN Specification - Business Process Model and Notation. [online] Available at: <http://bpmn.org> [Accessed 12 May 2017].
- Buyya, R., Yeo, C., Venugopal, S., Broberg, J. and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6).
- Chinosi, M. and Trombetta, A. (2012). BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 34(1).
- Dumas, M., La Rosa, M., Mendling, J. and Reijers, H. (2013). *Fundamentals of Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Eurostat (2016). *Cloud computing - statistics on the use by enterprises*. [online] Available at: http://ec.europa.eu/eurostat/statistics-explained/index.php/Cloud_computing_-_statistics_on_the_use_by_enterprises [Accessed 2 June 2017].
- Gartner (2007). *Introducing the Gartner IT Infrastructure and Operations Maturity Model*. [online] Available at: <http://www.ucipfg.com/Repositorio/MATI/MATI-05/BLOQUE-ACADEMICO/Unidad-4/lecturas/IntroducingtheGartnerITInfrastructureandoperations.pdf> [Accessed 7 Aug. 2017].
- Humphrey, W. (2008). *Managing the software process*. Boston, Mass.: Addison-Wesley.
- Kniberg, H. (2017). *Spotify engineering culture (part 1)*. [online] Labs. Available at: <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/> [Accessed 13 Sept. 2017].

- Kniberg, H. (2017). *Spotify engineering culture (part 2)*. [online] Labs. Available at: <https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/> [Accessed 13 Sept. 2017].
- Merlyn, V. (2017). *The Business-IT Maturity Model*. [online] Formicio.com. Available at: <http://formicio.com/index.php/archives/765> [Accessed 2 Aug. 2017].
- Park, S., Hwang, M., Lee, S. and Park, Y. (2015). A Generic Software Development Process Refined from Best Practices for Cloud Computing. *Sustainability*, 7(5).
- Pichler, R. (2011). *Agile product management with Scrum*. Upper Saddle River, NJ: Addison-Wesley.
- PROJECT MANAGEMENT INSTITUTE. (2017). *A guide to the project management body of knowledge (PMBOK guide)*. Newtown Square, Pa, Project Management Institute
- PROJECT MANAGEMENT INSTITUTE., Agile Alliance. (2017). *Agile Practice Guide*. Newtown Square, Pa, Project Management Institute.
- Salesforce/IDC (2016). *The Salesforce Economy: Enabling 1.9 Million New Jobs and \$389 Billion in New Revenue Over the Next Five Years*. [online] IDC. Available at: <http://www.salesforce.com/assets/pdf/misc/IDC-salesforce-economy-study-2016.pdf> [Accessed 1 Oct. 2017].
- Wang X., Edison H., Bajwa S.S., Giardino C., Abrahamsson P. (2016) *Key Challenges in Software Startups Across Life Cycle Stages*. In: Sharp H., Hall T. (eds) *Agile Processes, in Software Engineering, and Extreme Programming. XP 2016. Lecture Notes in Business Information Processing*, vol 251. Springer, Cham